Expectation Maximization

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

April 9th, 2007

Gaussian Bayes Classifier Reminder

$$P(y=i \mid \mathbf{x}_j) = \frac{p(\mathbf{x}_j \mid y=i)P(y=i)}{p(\mathbf{x}_j)}$$

Class mean

class covariance

$$P(y=i \mid \mathbf{x}_j) \propto \underbrace{\frac{1}{(2\pi)^{m/2} \|\Sigma_i\|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathbf{x}_j - \mu_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}_j - \mu_i\right)\right]}_{\text{Gaussian likelihood}} \underbrace{P(y=i)}_{\text{prior}}$$

Next… back to Density Estimation

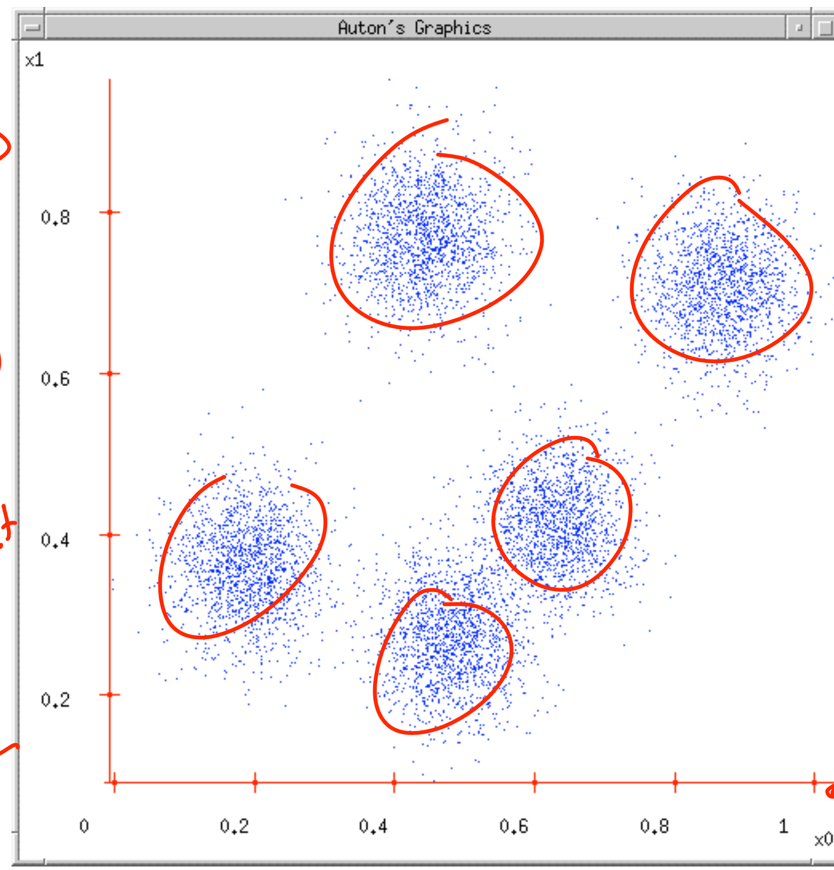
What if we want to do density estimation with multimodal or clumpy data?

want to represent $P(X)$

$$P(x) = \sum_i P(X \mid y=i) \cdot P(y=i)$$

Gaussian

weight



$X = \{x_1, x_2\}$

Marginal likelihood for general case

$$P(y=i \mid \mathbf{x}_j) \propto \frac{1}{(2\pi)^{m/2} \|\Sigma_i\|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathbf{x}_j-\mu_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}_j-\mu_i\right)\right] P(y=i)$$

defn. of GMM

$x_j \leftarrow$ observed

assumption : $x_j \sim$ GMM

$P(x_j) = \sum_i P(y=i) \cdot P(x_j \mid y_j=i)$

don't observe y_j

$\Rightarrow$ max $P(x_j)$

Gaussian

- Marginal likelihood:

log

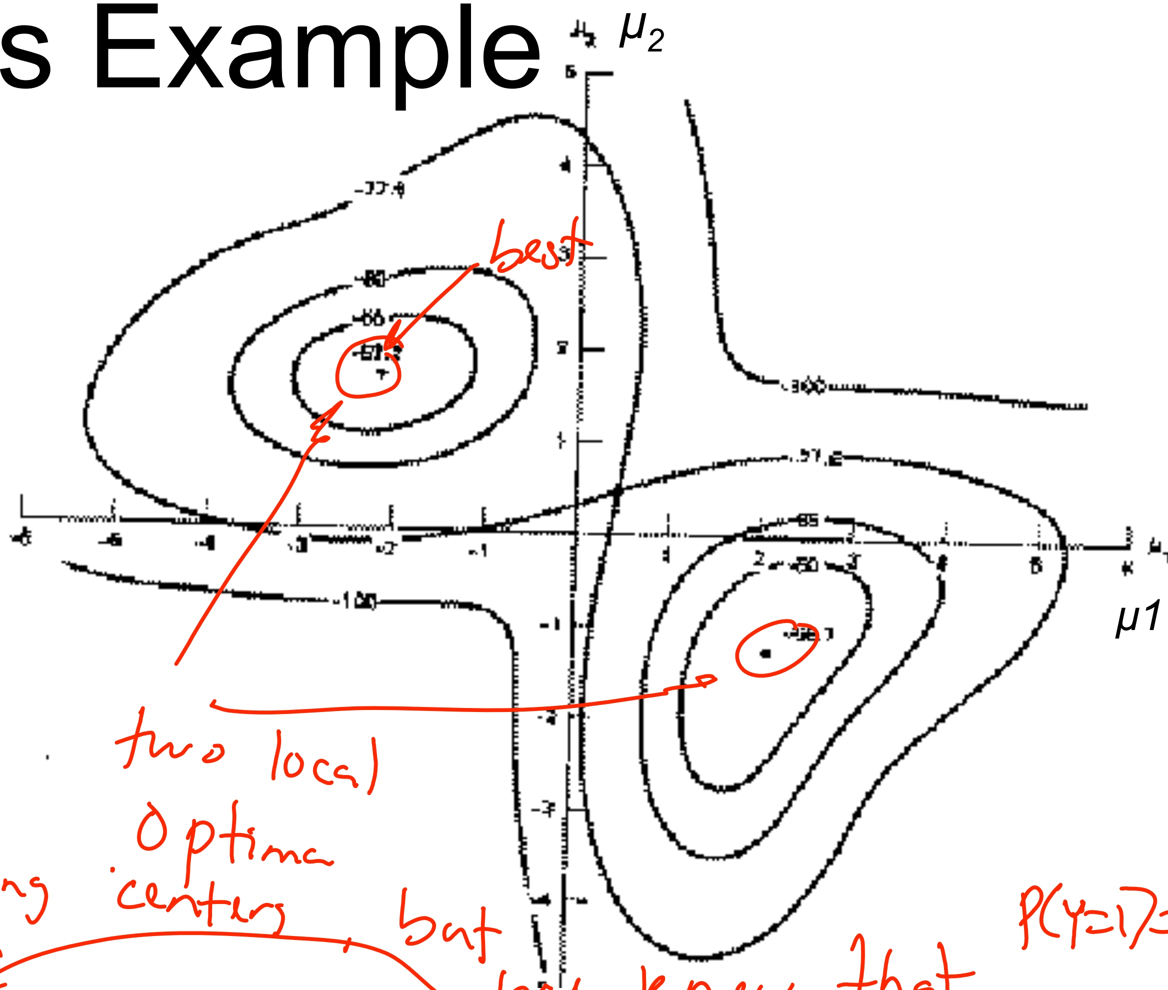
$$\prod_{j=1}^{m} P(\mathbf{x}_j) = \log \prod_{j=1}^{m} \sum_{i=1}^{k} P(\mathbf{x}_j, y=i)$$

$$= \log \prod_{j=1}^{m} \sum_{i=1}^{k} \frac{1}{(2\pi)^{m/2} \|\Sigma_i\|^{1/2}} \exp\left[-\frac{1}{2}\left(\mathbf{x}_j-\mu_i\right)^T \Sigma_i^{-1}\left(\mathbf{x}_j-\mu_i\right)\right] P(y=i)$$

$$= \sum_{j=1}^{m} \log \sum_{i=1}^{k}$$

Duda & Hart's Example

$\max_{\mu} P(X \mid \mu)$



Graph of

$\log P(x_1, x_2 .. x_{25} \mid \mu_1, \mu_2)$

against μ_1 (→) and μ_2 (↑)

flipping centers, but we knew that

Max likelihood = (μ_1 =-2.13, μ_2 =1.668)

Local minimum, but very close to global at (μ_1 =2.085, μ_2 =-1.257)*

$P(Y=1)=2/3$

$P(Y=2)=1/3$

* corresponds to switching y_1 with y_2.

Finding the max likelihood $\mu_1,\mu_2..\mu_k$

We can compute P(data | $\boldsymbol{\mu}_1,\boldsymbol{\mu}_2..\boldsymbol{\mu}_k$)

How do we find the $\boldsymbol{\mu}_i$'s which give max. likelihood?

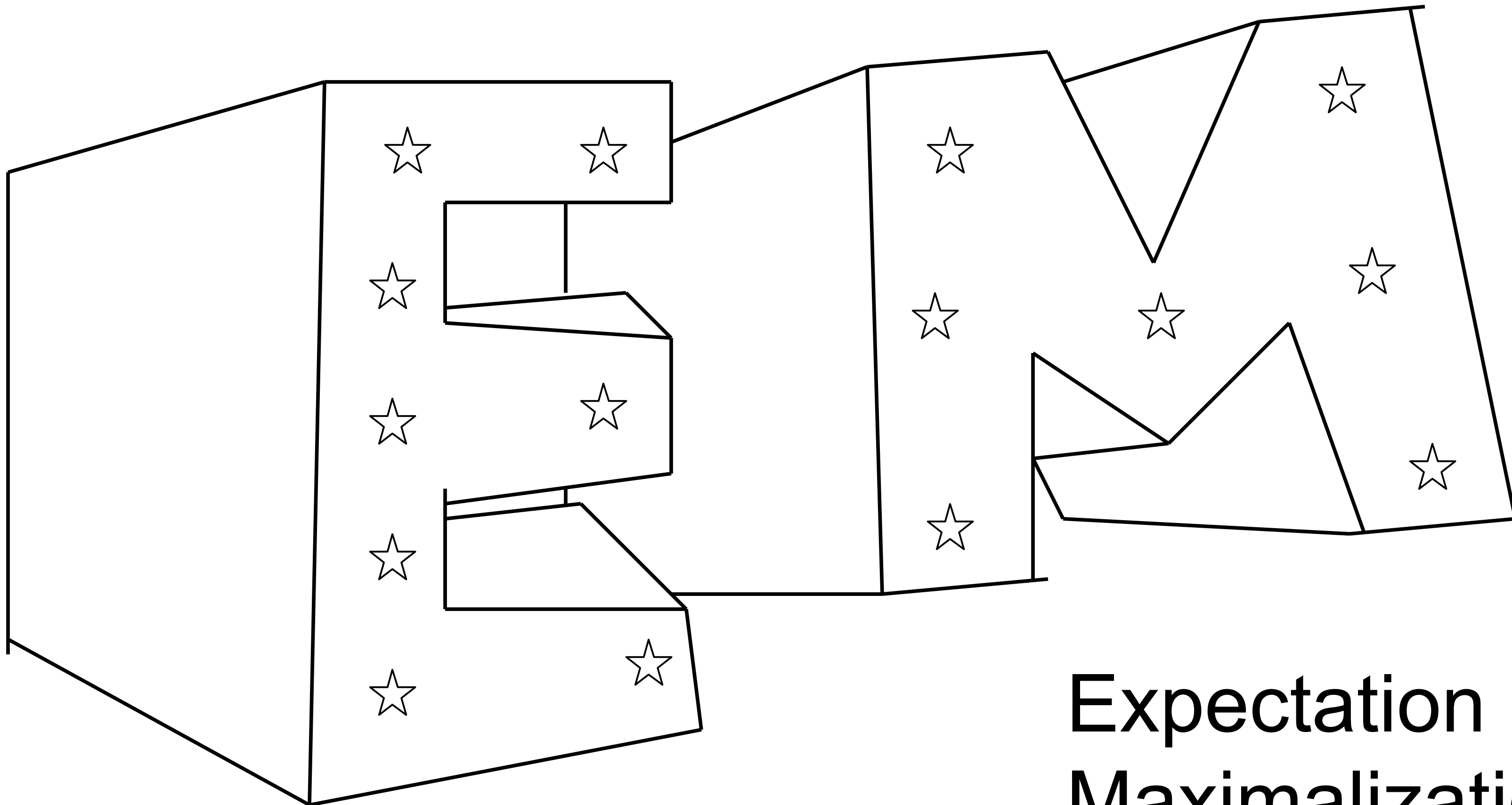
- The normal max likelihood trick:

  Set $\frac{\partial}{\partial \mu_i}$ log Prob (….) = 0

  and solve for μ_i's.

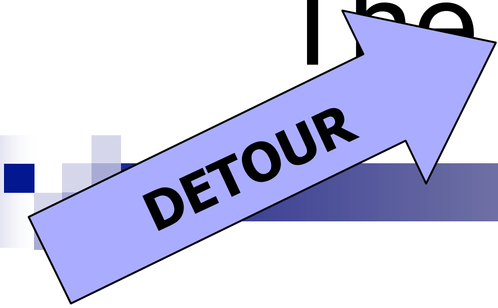
  # Here you get non-linear non-analytically-solvable equations
- Use gradient descent

  Slow but doable
- Use a much faster, cuter, and recently very popular method…



Expectation
Maximalization

The E.M. Algorithm



- We'll get back to unsupervised learning soon
- But now we'll look at an even simpler case with hidden information
- The EM algorithm
  - Can do trivial things, such as the contents of the next few slides
  - An excellent way of doing our unsupervised learning problem, as we'll see
  - Many, many other uses, including learning BNs with hidden data

Silly Example

Let events be "grades in a class"

w_1 = Gets an A P(A) = ½

w_2 = Gets a B P(B) = μ

w_3 = Gets a C P(C) = 2μ

w_4 = Gets a D P(D) = ½-3μ

(Note $0 \leq \mu \leq 1/6$)

Assume we want to estimate μ from data. In a given class there were

a A's
b B's
c C's
d D's

What's the maximum likelihood estimate of μ given a,b,c,d ?

Trivial Statistics

$P(A) = \frac{1}{2}$ $P(B) = \mu$ $P(C) = 2\mu$ $P(D) = \frac{1}{2}-3\mu$

$P(a,b,c,d \mid \mu) = K(\frac{1}{2})^a(\mu)^b(2\mu)^c(\frac{1}{2}-3\mu)^d$

$\log P(a,b,c,d \mid \mu) = \log K + a\log \frac{1}{2} + b\log \mu + c\log 2\mu + d\log (\frac{1}{2}-3\mu)$

$$\text{FOR MAX LIKE } \mu, \text{ SET } \frac{\partial \text{LogP}}{\partial \mu} = 0$$

$$\frac{\partial \text{LogP}}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0$$

$$\text{Gives max like } \mu = \frac{b+c}{6(b+c+d)}$$

So if class got

A	B	C	D
14	6	9	10

$$\text{Max like } \mu = \frac{1}{10}$$

Boring, but true!

Same Problem with Hidden Information

REMEMBER

$P(A) = \frac{1}{2}$

$P(B) = \mu$

$P(C) = 2\mu$

$P(D) = \frac{1}{2} - 3\mu$

Someone tells us that

Number of High grades (A's + B's) = h

Number of C's = c

Number of D's = d

What is the max. like estimate of μ now?

Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) = h

Number of C's = c

Number of D's = d

REMEMBER

P(A) = ½

P(B) = μ

P(C) = 2μ

P(D) = ½-3μ

What is the max. like estimate of μ now?

We can answer this question circularly:

EXPECTATION

If we know the value of μ we could compute the expected value of *a* and *b*

Since the ratio a:b should be the same as the ratio ½ : μ

$$a = \frac{\frac{1}{2}}{\frac{1}{2} + \mu} h \qquad b = \frac{\mu}{\frac{1}{2} + \mu} h$$

MAXIMIZATION

If we know the expected values of *a* and *b* we could compute the maximum likelihood value of μ

$$\mu = \frac{b + c}{6(b + c + d)}$$

E.M. for our Trivial Problem

REMEMBER

P(A) = ½

P(B) = μ

P(C) = 2μ

P(D) = ½-3μ

We begin with a guess for μ

We iterate between EXPECTATION and MAXIMALIZATION to improve our estimates of μ and *a* and *b*.

Define $\mu^{(t)}$ the estimate of μ on the t'th iteration

$b^{(t)}$ the estimate of *b* on t'th iteration

$$\mu^{(0)} = \text{initial guess}$$

$$b^{(t)} = \frac{\mu^{(t)}h}{\frac{1}{2} + \mu^{(t)}} = \mathrm{E}\left[b \mid \mu^{(t)}\right]$$

E-step

$$\mu^{(t+1)} = \frac{b^{(t)} + c}{6\left(b^{(t)} + c + d\right)}$$

$$= \text{max like est. of } \mu \text{ given } b^{(t)}$$

M-step

Continue iterating until converged.
Good news: Converging to local optimum is assured.
Bad news: I said "local" optimum.

E.M. Convergence

- Convergence proof based on fact that Prob(data | μ) must increase or remain same between each iteration [NOT OBVIOUS]
- But it can never exceed 1 [OBVIOUS]

So it must therefore converge [OBVIOUS]

In our example, suppose we had

$h = 20$

$c = 10$

$d = 10$

$\mu^{(0)} = 0$

t	$\mu^{(t)}$	$b^{(t)}$
0	0	0
1	0.0833	2.857
2	0.0937	3.158
3	0.0947	3.185
4	0.0948	3.187
5	0.0948	3.187
6	0.0948	3.187

Convergence is generally <u>linear</u>: error decreases by a constant factor each time step.

Back to Unsupervised Learning of GMMs – a simple case

A simple case:

We have unlabeled data $\boldsymbol{x}_1$ $\boldsymbol{x}_2$ … $\boldsymbol{x}_m$

We know there are k classes

We know $P(y_1)$ $P(y_2)$ $P(y_3)$ … $P(y_k)$

We <u>don't</u> know $\boldsymbol{\mu}_1$ $\boldsymbol{\mu}_2$ .. $\boldsymbol{\mu}_k$

We can write P(data | $\boldsymbol{\mu}_1$…. $\boldsymbol{\mu}_k$)

$$= \mathrm{p}\left(x_1 \ldots x_m \middle| \mu_1 \ldots \mu_k\right)$$

$$= \prod_{j=1}^{m} \mathrm{p}\left(x_j \middle| \mu_1 \ldots \mu_k\right)$$

$$= \prod_{j=1}^{m} \sum_{i=1}^{k} \mathrm{p}\left(x_j \middle| \mu_i\right) \mathrm{P}\left(y = i\right)$$

$$\propto \prod_{j=1}^{m} \sum_{i=1}^{k} \exp\left(-\frac{1}{2\sigma^2}\left\|x_j - \mu_i\right\|^2\right) \mathrm{P}\left(y = i\right)$$

EM for simple case of GMMs: The E-step

- If we know $\mu_1,\ldots,\mu_k$ $\rightarrow$ easily compute prob. point x_j belongs to class y=i

$$p\left(y=i \middle| x_j,\mu_1\ldots\mu_k\right) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|x_j-\mu_i\right\|^2\right)P\left(y=i\right)$$

EM for simple case of GMMs: The M-step

- If we know prob. point x_j belongs to class $y=i$

  $\rightarrow$ MLE for μ_i is weighted average

  - imagine k copies of each x_j, each with weight $P(y=i|x_j)$:

$$\mu_i = \frac{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right) x_j}{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right)}$$

E.M. for GMMs

E-step

Compute "expected" classes of all datapoints for each class

$$\mathrm{p}\left(y=i \middle| x_j, \mu_1 \ldots \mu_k\right) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|x_j - \mu_i\right\|^2\right)\mathrm{P}(y=i)$$

Just evaluate a Gaussian at x_j

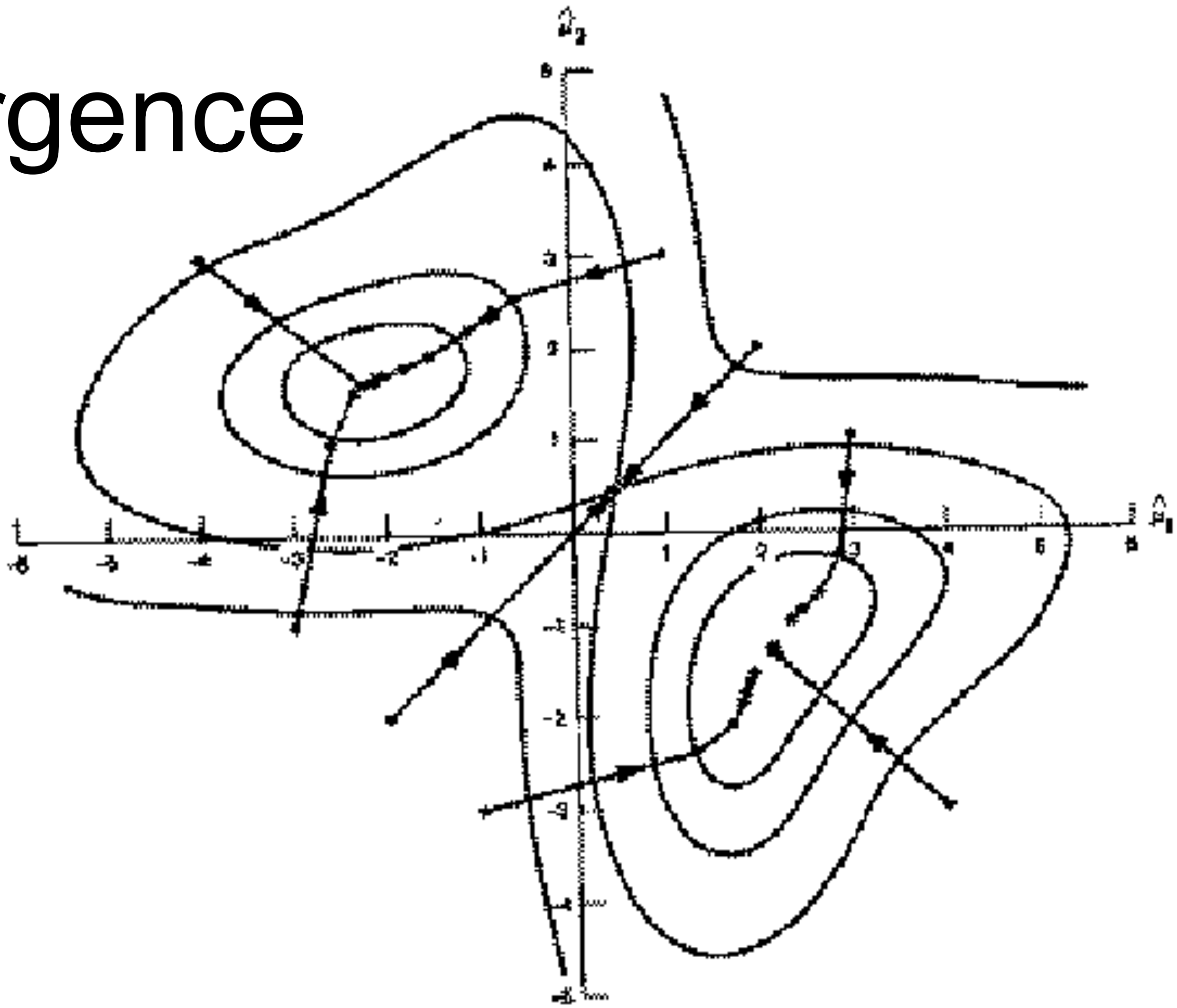
M-step

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i = \frac{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right) x_j}{\sum_{j=1}^{m} P\left(y=i \middle| x_j\right)}$$

E.M. Convergence

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. ! convergence to a local optimum guaranteed
- See Neal & Hinton reading on class webpage



- This algorithm is REALLY USED. And in high dimensional state spaces, too. E.G. Vector Quantization for Speech Data

E.M. for General GMMs

$p_i^{(t)}$ is shorthand for estimate of $P(y=i)$ on t'th iteration

Iterate. On the *t*'th iteration let our estimates be

$$\lambda_t = \{ \mu_1^{(t)}, \mu_2^{(t)} \ldots \mu_k^{(t)}, \Sigma_1^{(t)}, \Sigma_2^{(t)} \ldots \Sigma_k^{(t)}, p_1^{(t)}, p_2^{(t)} \ldots p_k^{(t)} \}$$

E-step

Compute "expected" classes of all datapoints for each class

$$P(y = i | x_j, \lambda_t) \propto p_i^{(t)} p(x_j | \mu_i^{(t)}, \Sigma_i^{(t)})$$

Just evaluate a Gaussian at x_j

M-step

Compute Max. like μ given our data's class membership distributions

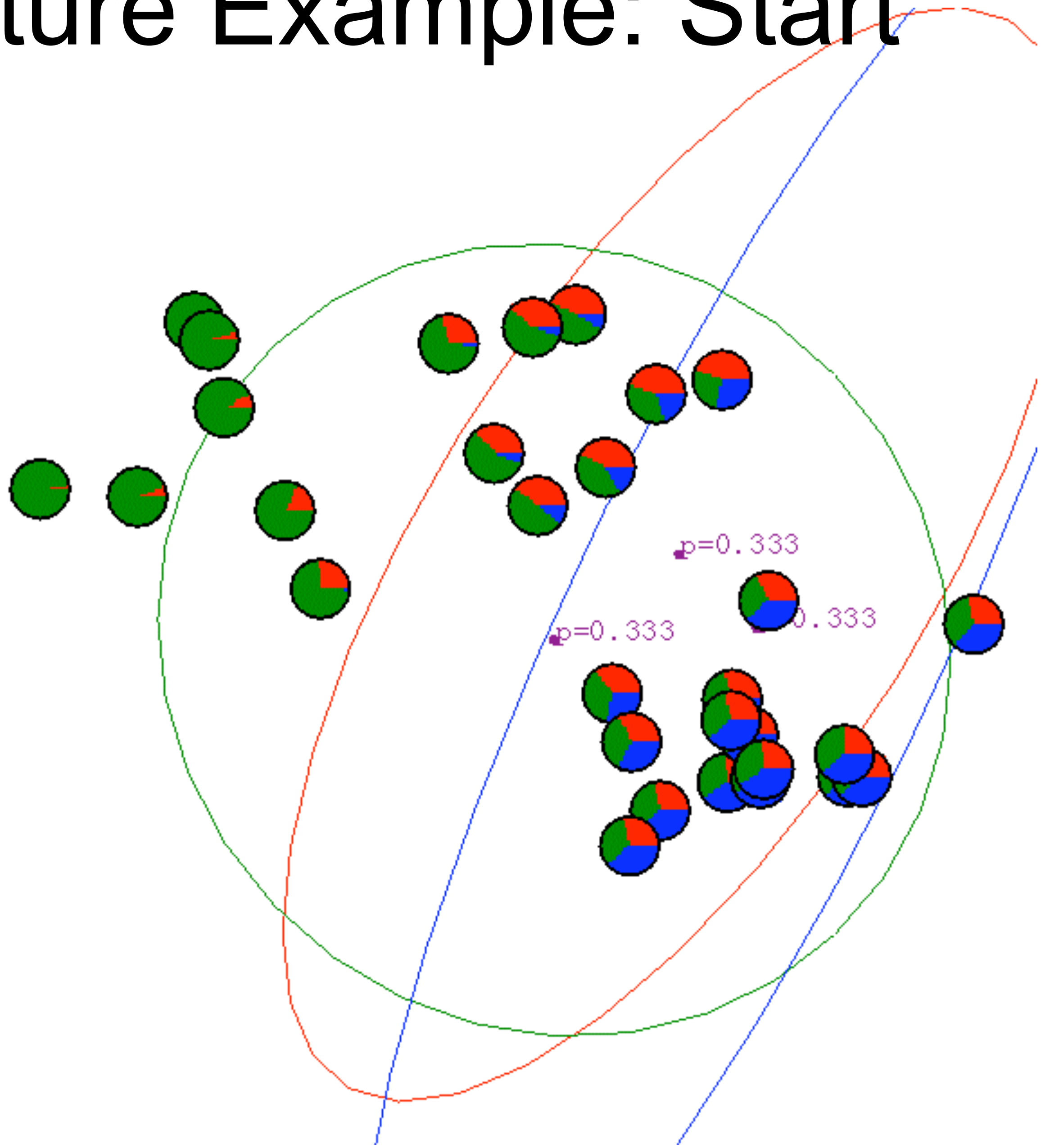
$$\mu_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t) x_j}{\sum_j P(y = i | x_j, \lambda_t)}$$

$$\Sigma_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t) \left[x_j - \mu_i^{(t+1)}\right]\left[x_j - \mu_i^{(t+1)}\right]^T}{\sum_j P(y = i | x_j, \lambda_t)}$$

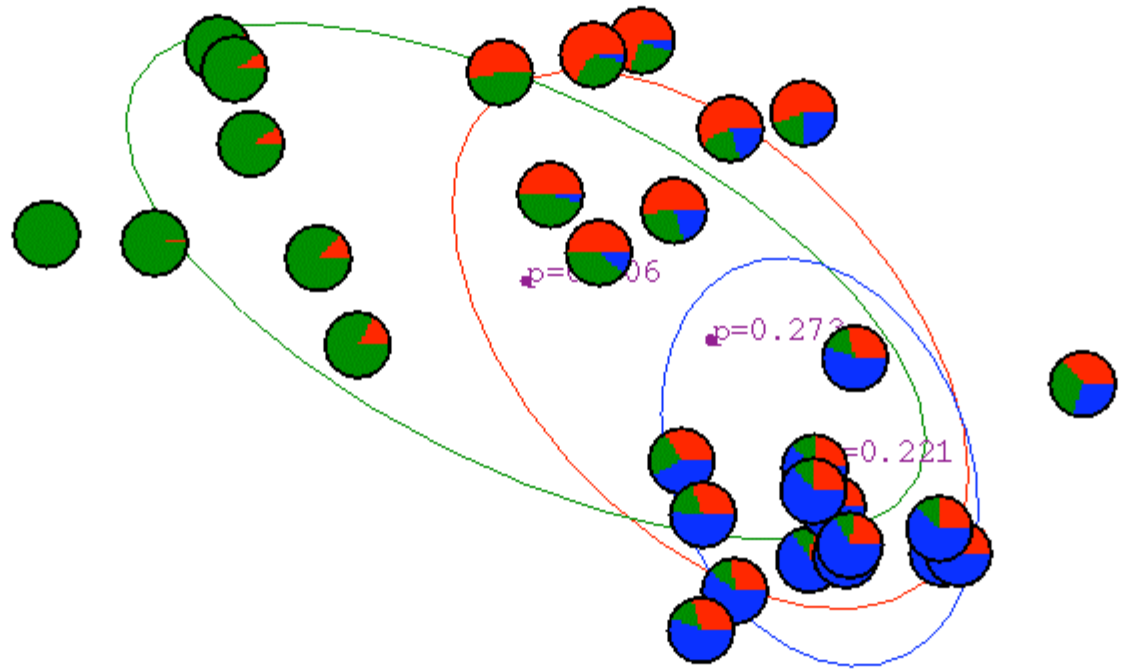
$$p_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t)}{m}$$

m = #records

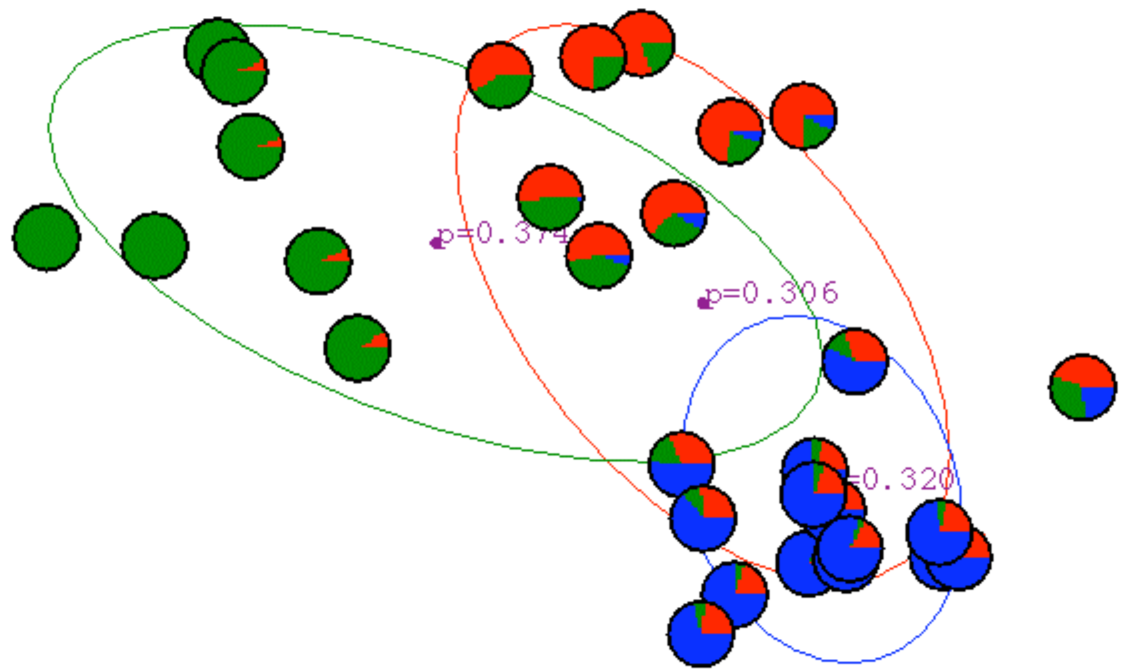
Gaussian Mixture Example: Start



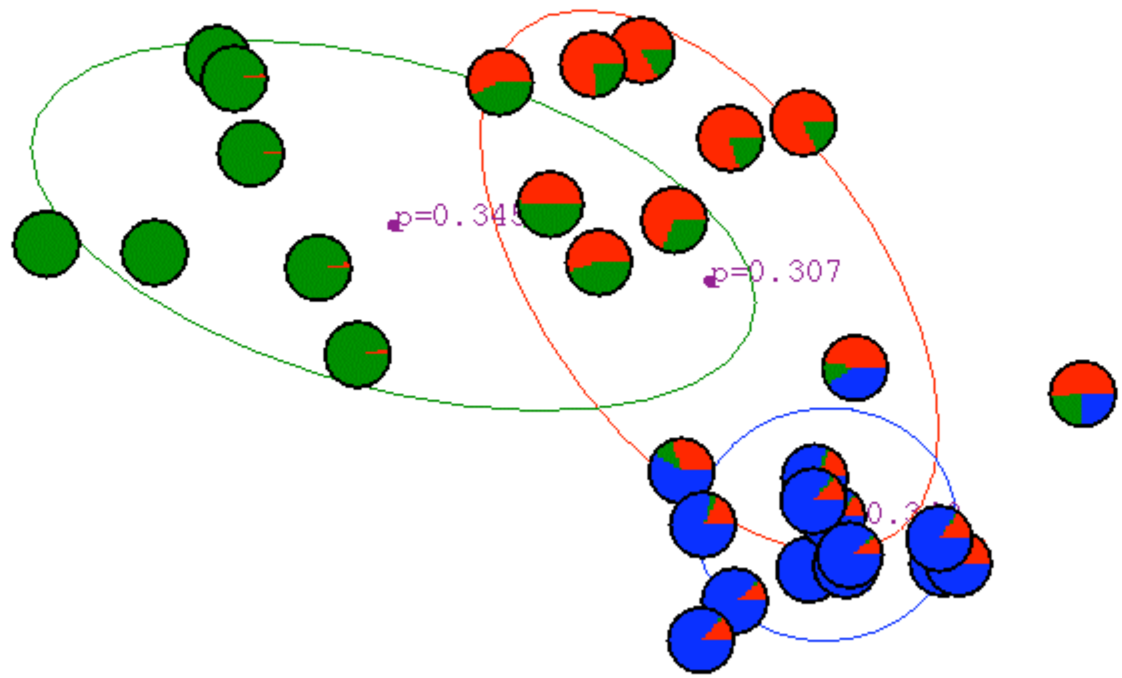
After first iteration



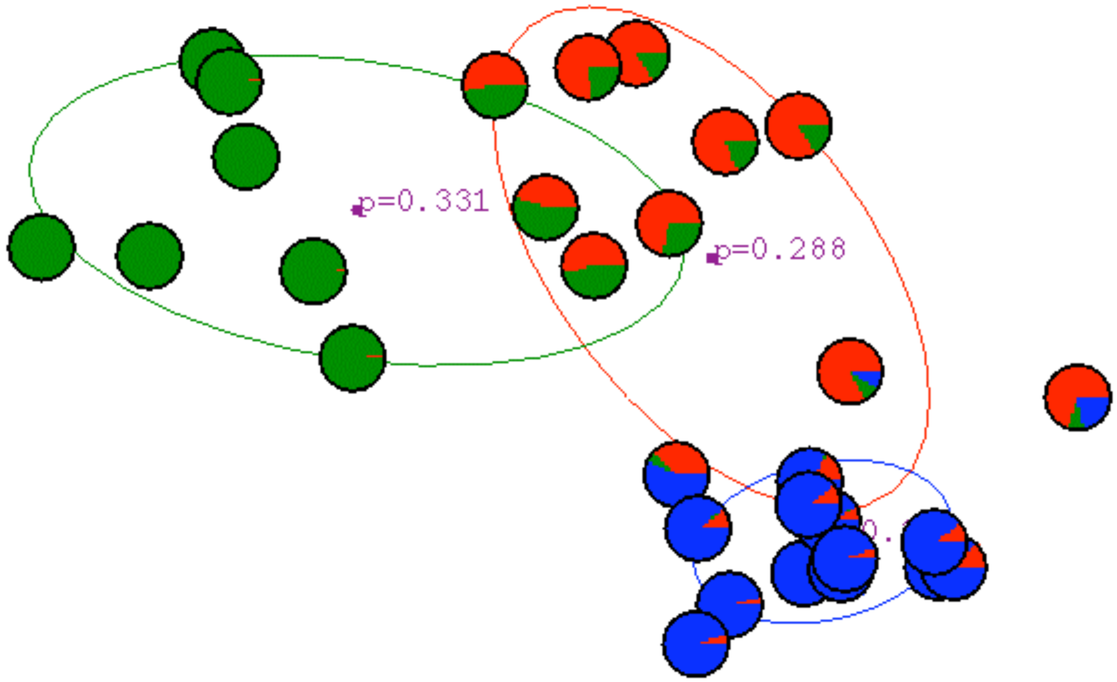
After 2nd iteration



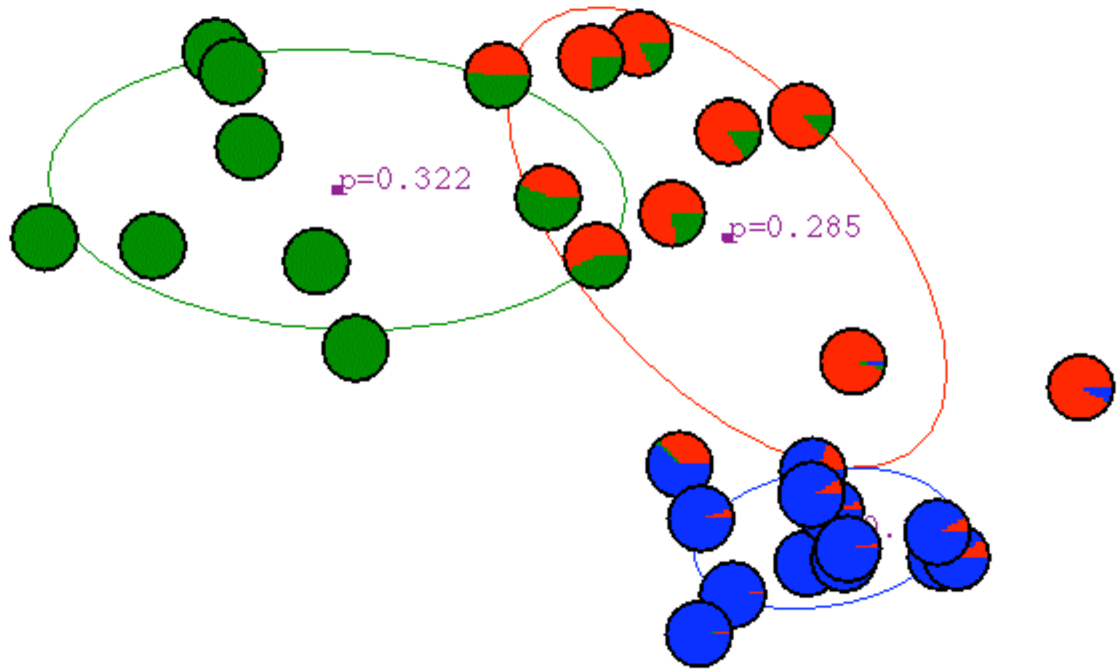
After 3rd iteration



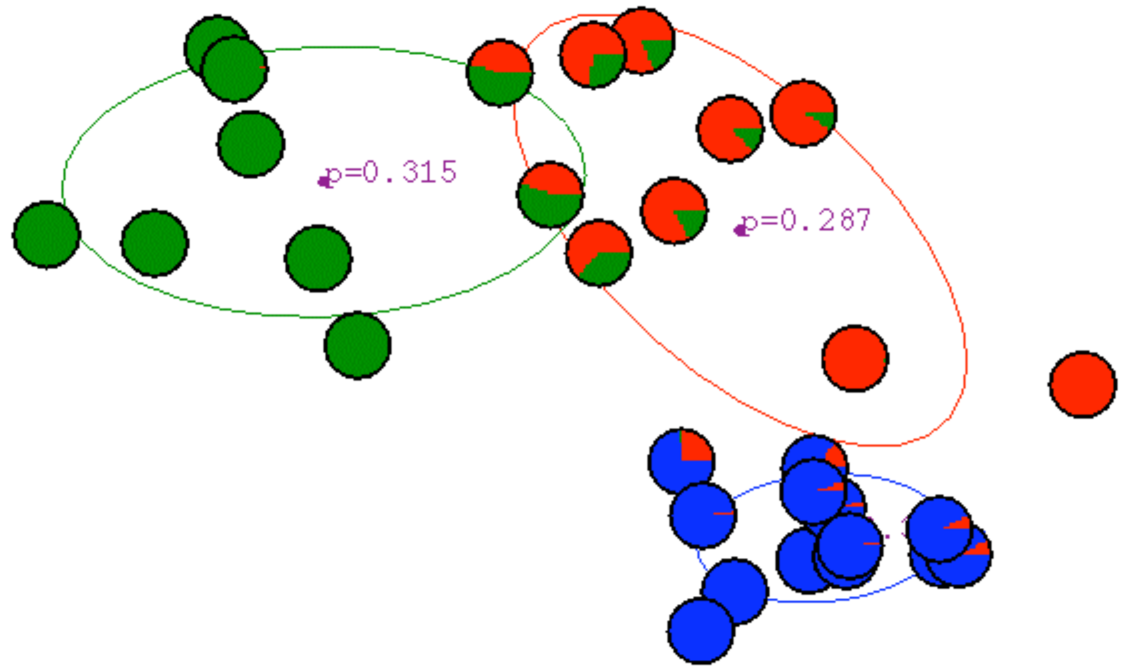
After 4th iteration



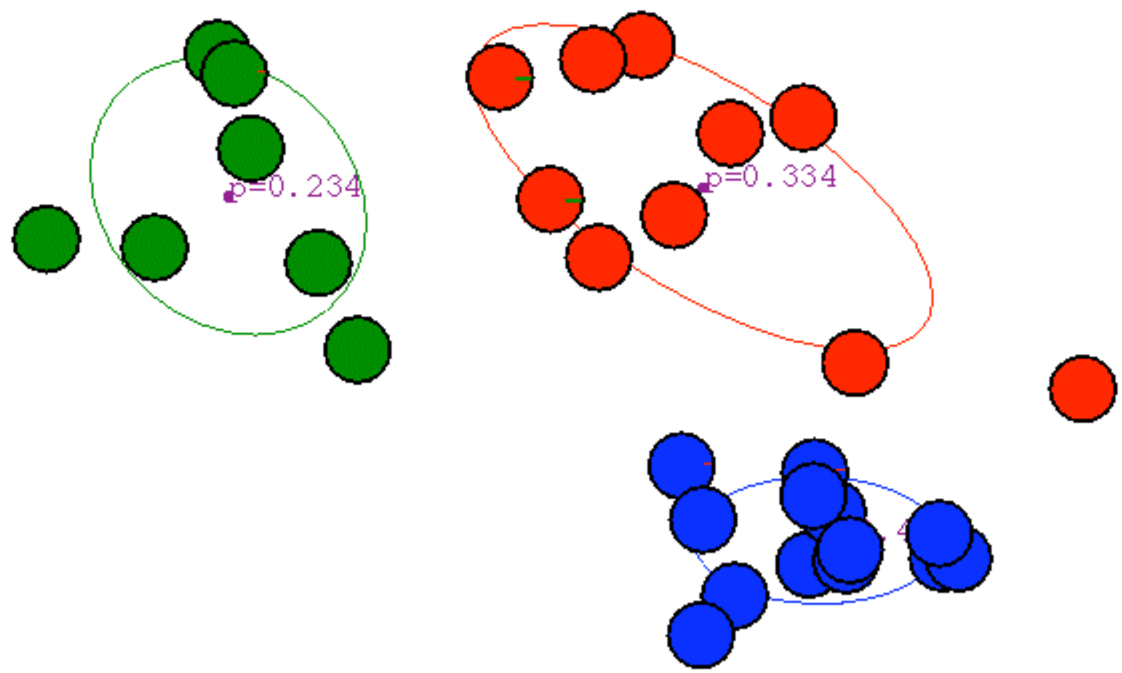
After 5th iteration



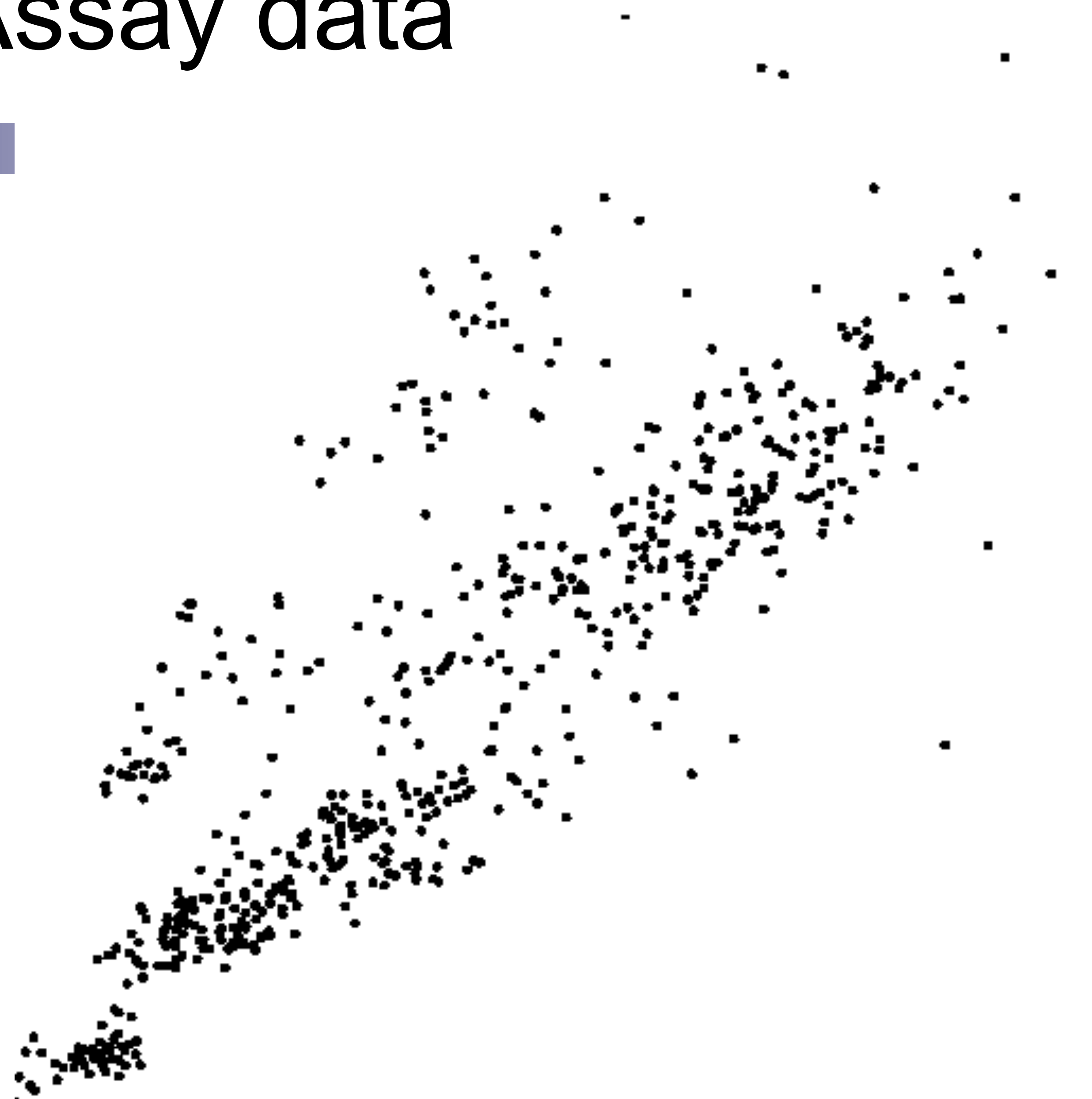
After 6th iteration



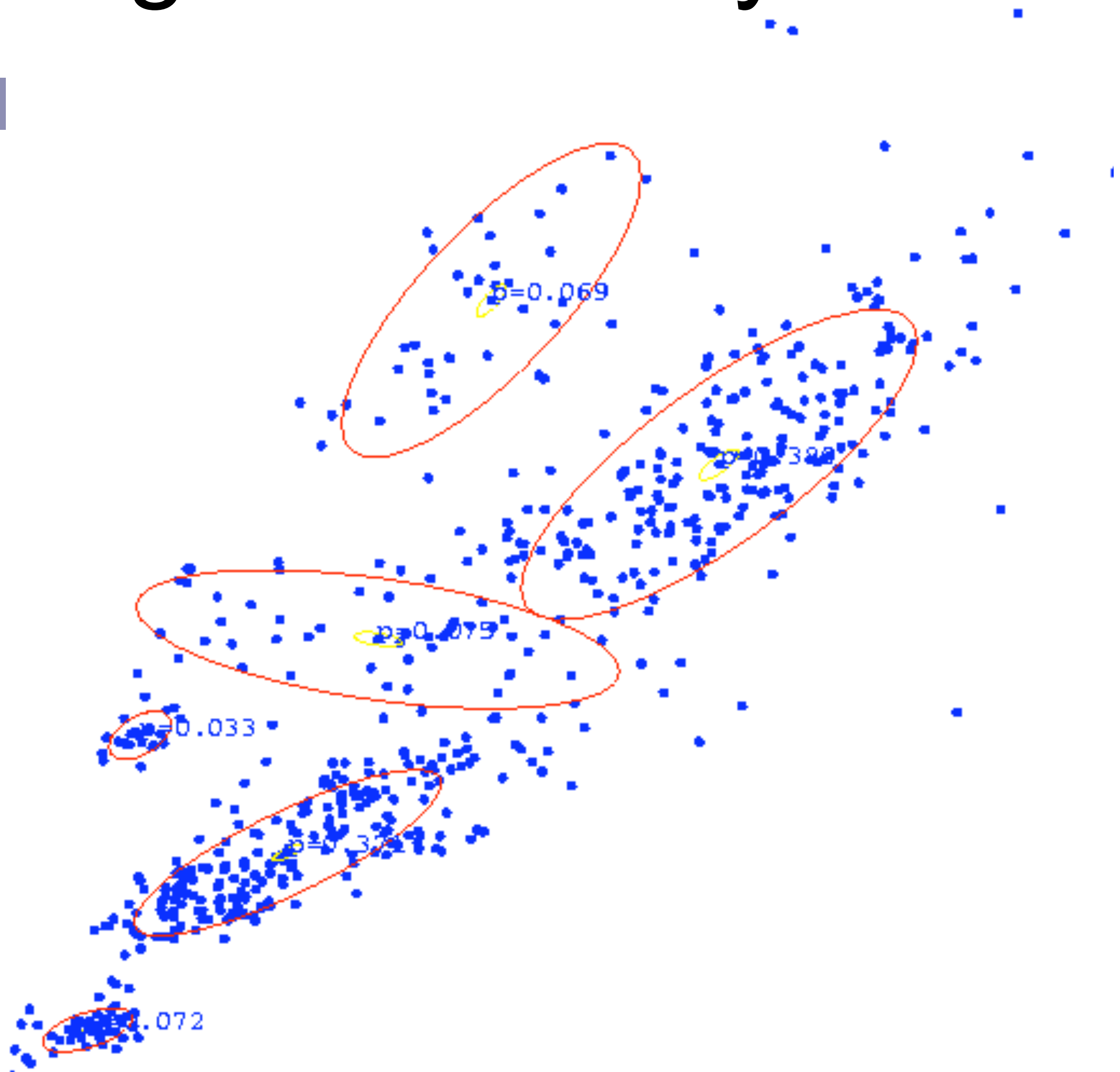
After 20th iteration



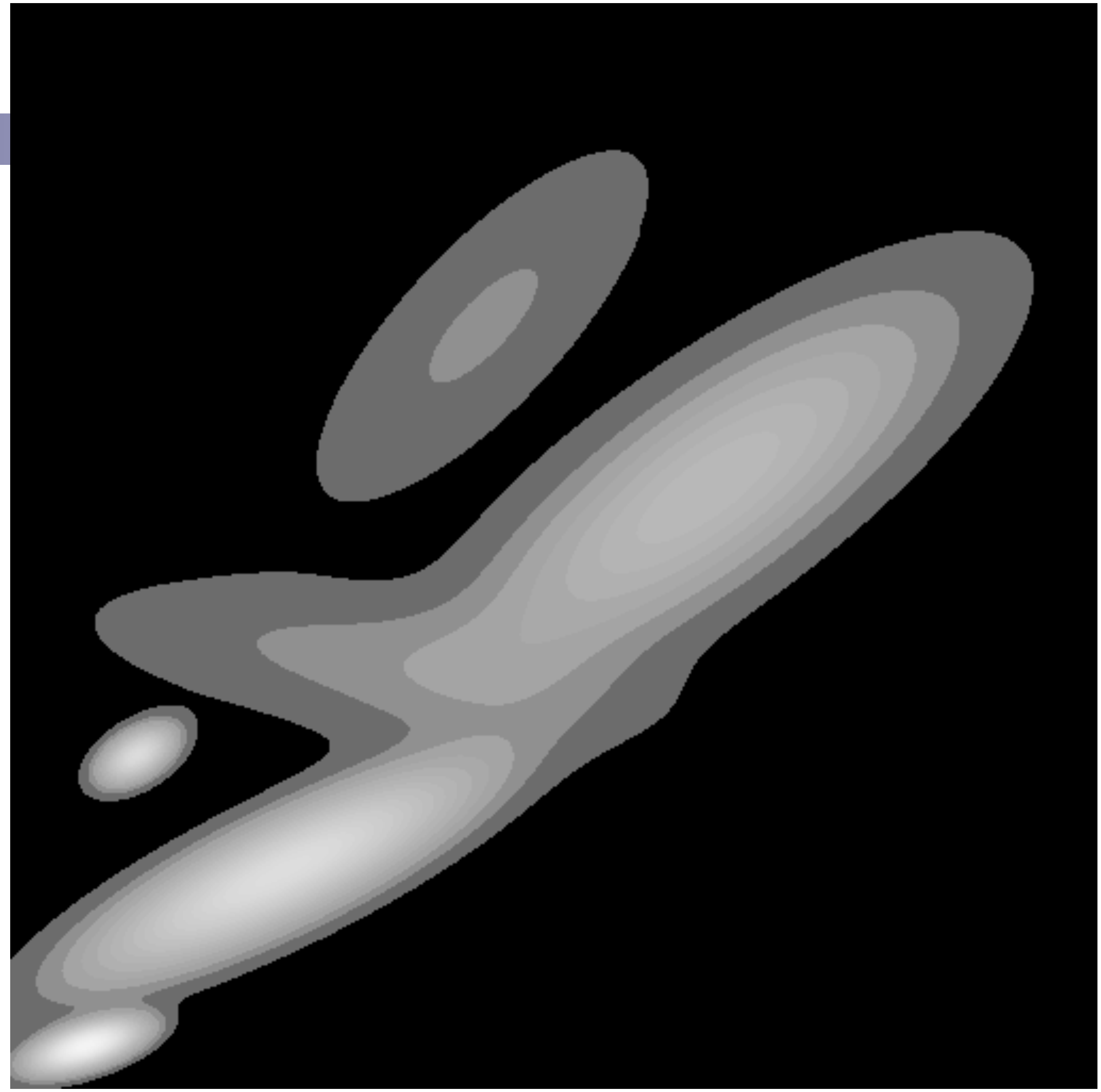
Some Bio Assay data



GMM clustering of the assay data

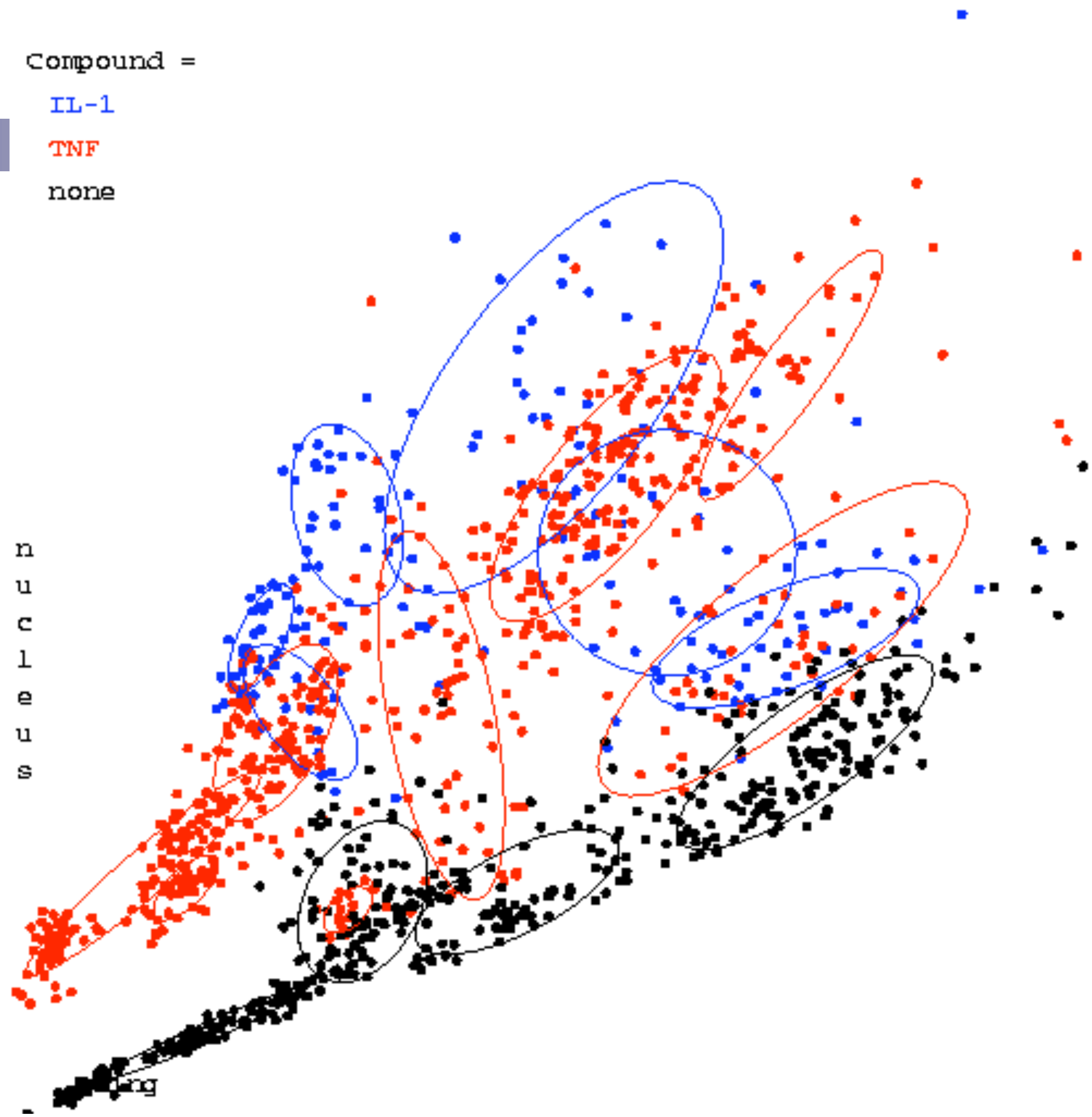


Resulting Density Estimator

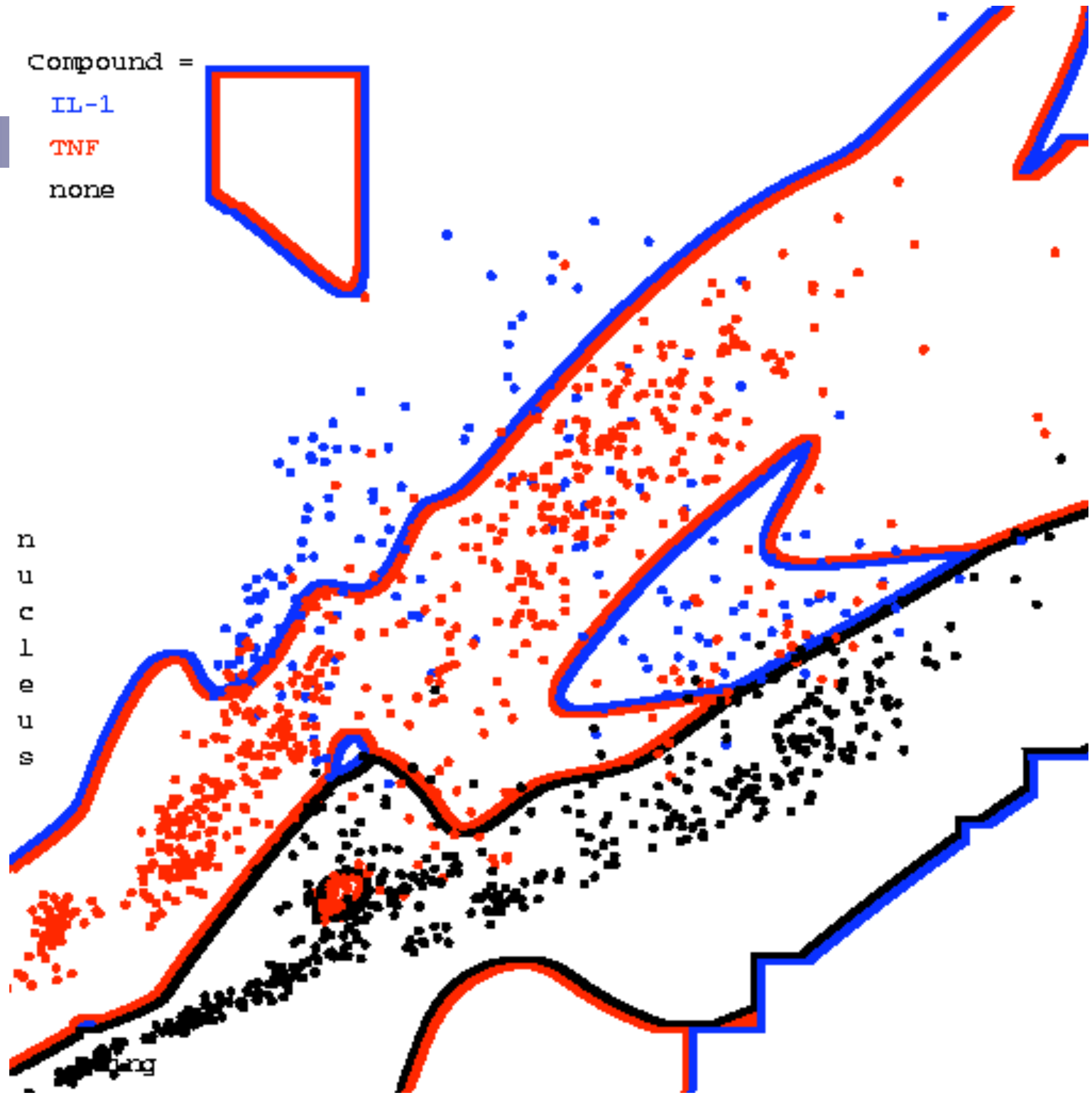


Three classes of assay

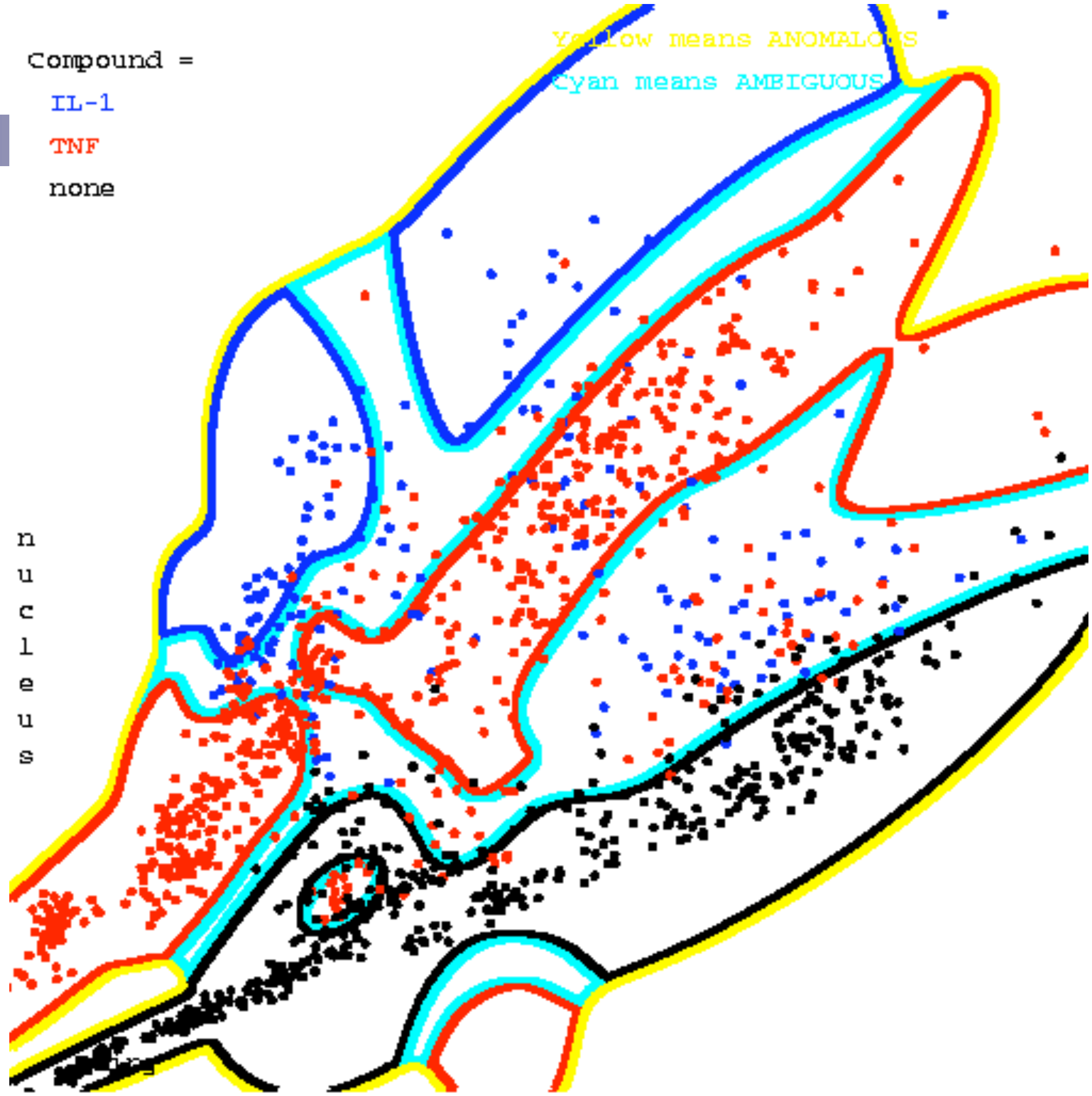
(each learned with it's own mixture model)



Resulting Bayes Classifier



Resulting Bayes Classifier, using posterior probabilities to alert about ambiguity and anomalousness



Yellow means anomalous

Cyan means ambiguous

The general learning problem with missing data

- Marginal likelihood – **x** is observed, **z** is missing:

$$
\begin{aligned}
\ell(\theta : \mathcal{D}) &= \log \prod_{j=1}^{m} P(\mathbf{x}_j \mid \theta) \\
&= \sum_{j=1}^{m} \log P(\mathbf{x}_j \mid \theta) \\
&= \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)
\end{aligned}
$$

E-step

- **x** is observed, **z** is missing
- Compute probability of missing data given current choice of θ
  - $Q(\mathbf{z}|\mathbf{x}_j)$ for each $\mathbf{x}_j$
    - e.g., probability computed during classification step
    - corresponds to "classification step" in K-means

$$Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \;=\; P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})$$

Jensen's inequality

$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{z} \mid \mathbf{x}_j) P(\mathbf{x}_j \mid \theta)$$

- **Theorem**: log $\sum_{\mathbf{z}}$ P(**z**) f(**z**) $\geq$ $\sum_{\mathbf{z}}$ P(**z**) log f(**z**)

Applying Jensen's inequality

- Use: $\log \sum_{\mathbf{z}} P(\mathbf{z})\, f(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log f(\mathbf{z})$

$$\ell(\theta^{(t)} : \mathcal{D}) \;=\; \sum_{j=1}^{m} \log \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)})}{Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j)}$$

The M-step maximizes lower bound on weighted data

- Lower bound from Jensen's:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)}) + m.H(Q^{(t+1)})$$

- Corresponds to weighted dataset:
  - $<\mathbf{x}_1, \mathbf{z}=1>$ with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}_1)$
  - $<\mathbf{x}_1, \mathbf{z}=2>$ with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}_1)$
  - $<\mathbf{x}_1, \mathbf{z}=3>$ with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}_1)$
  - $<\mathbf{x}_2, \mathbf{z}=1>$ with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}_2)$
  - $<\mathbf{x}_2, \mathbf{z}=2>$ with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}_2)$
  - $<\mathbf{x}_2, \mathbf{z}=3>$ with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}_2)$
  - …

The M-step

$$\ell(\theta^{(t)} : \mathcal{D}) \geq \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)}) + m.H(Q^{(t+1)})$$

- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta)$$

- Use expected counts instead of counts:
  - If learning requires Count(**x**,**z**)
  - Use $E_{Q(t+1)}$[Count(**x**,**z**)]

Convergence of EM

- Define potential function F(θ,Q):

$$\ell(\theta:\mathcal{D}) \;\geq\; F(\theta,Q) = \sum_{j=1}^{m}\sum_{\mathbf{z}} Q(\mathbf{z}\mid\mathbf{x}_j)\log\frac{P(\mathbf{z},\mathbf{x}_j\mid\theta)}{Q(\mathbf{z}\mid\mathbf{x}_j)}$$

- EM corresponds to coordinate ascent on F
  - Thus, maximizes lower bound on marginal log likelihood

M-step is easy

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta)$$

- Using potential function

$$F(\theta, Q^{(t+1)}) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) + m.H(Q^{(t+1)})$$

E-step also doesn't decrease potential function 1

- Fixing θ to $\theta^{(t)}$:

$$\ell(\theta^{(t)} : \mathcal{D}) \quad \geq \quad F(\theta^{(t)}, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

KL-divergence

- Measures distance between distributions

$$KL(Q||P) = \sum_z Q(z) \log \frac{Q(z)}{P(z)}$$

- KL=zero if and only if Q=P

E-step also doesn't decrease potential function 2

- Fixing θ to $\theta^{(t)}$:

$$
\begin{aligned}
\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) &= \ell(\theta^{(t)} : \mathcal{D}) + \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)} \\
&= \ell(\theta^{(t)} : \mathcal{D}) - m \sum_{j=1}^{m} KL\left(Q(\mathbf{z} \mid \mathbf{x}_j) || P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\right)
\end{aligned}
$$

E-step also doesn't decrease potential function 3

$$\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) \;=\; \ell(\theta^{(t)} : \mathcal{D}) - m \sum_{j=1}^{m} KL\left(Q(\mathbf{z} \mid \mathbf{x}_j) || P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\right)$$

- Fixing θ to $\theta^{(t)}$
- Maximizing $F(\theta^{(t)},Q)$ over Q → set Q to posterior probability:

$$Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \;\leftarrow\; P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})$$

- Note that

$$F(\theta^{(t)}, Q^{(t+1)}) \;=\; \ell(\theta^{(t)} : \mathcal{D})$$

EM is coordinate ascent

$$\ell(\theta : \mathcal{D}) \;\;\geq\;\; F(\theta, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

- **M-step**: Fix Q, maximize F over θ (a lower bound on $\ell(\theta : \mathcal{D})$):

$$\ell(\theta : \mathcal{D}) \;\;\geq\;\; F(\theta, Q^{(t)}) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) + m.H(Q^{(t)})$$

- **E-step**: Fix θ, maximize F over Q:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) \;\;=\;\; \ell(\theta^{(t)} : \mathcal{D}) - m \sum_{j=1}^{m} KL\left(Q(\mathbf{z} \mid \mathbf{x}_j) || P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\right)$$

  - "Realigns" F with likelihood:

$$F(\theta^{(t)}, Q^{(t+1)}) \;\;=\;\; \ell(\theta^{(t)} : \mathcal{D})$$

What you should know

- K-means for clustering:
  - algorithm
  - converges because it's coordinate ascent
- EM for mixture of Gaussians:
  - How to "learn" maximum likelihood parameters (locally max. like.) in the case of unlabeled data
- Be happy with this kind of probabilistic analysis
- Remember, E.M. can get stuck in local minima, and empirically it <u>DOES</u>
- EM is coordinate ascent
- General case for EM

Acknowledgements

- K-means & Gaussian mixture models presentation contains material from excellent tutorial by Andrew Moore:
  - http://www.autonlab.org/tutorials/
- K-means Applet:
  - http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html
- Gaussian mixture models Applet:
  - http://www.neurosci.aist.go.jp/%7Eakaho/MixtureEM.html

EM for HMMs a.k.a. The Baum-Welch Algorithm

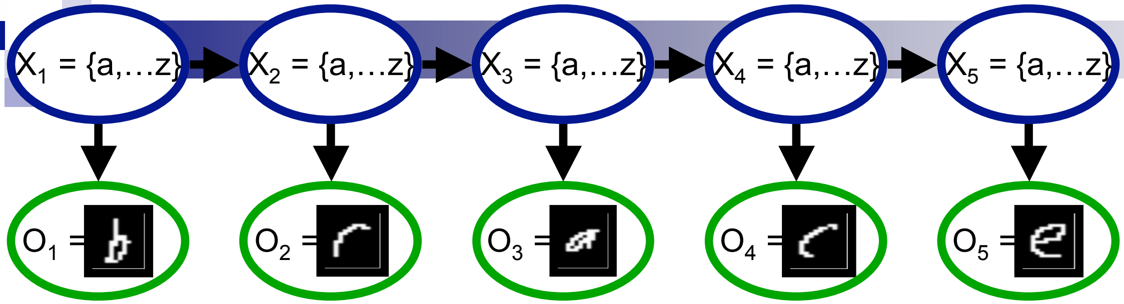
Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

April 9th, 2007

Learning HMMs from fully observable data is easy



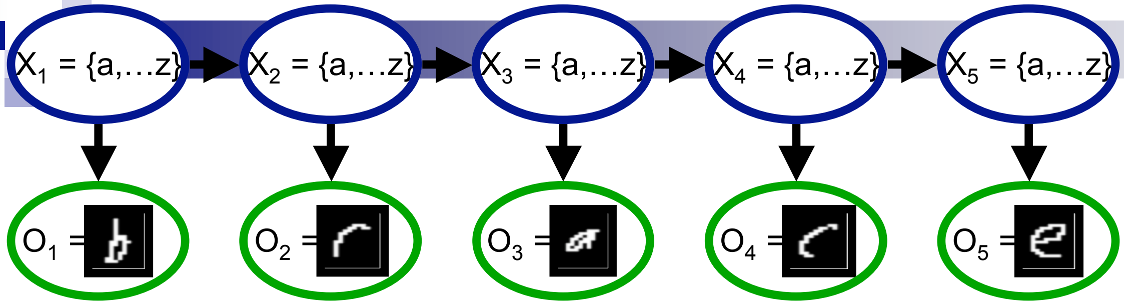
Learn 3 distributions:

$$P(X_1)$$

$$P(O_i \mid X_i)$$

$$P(X_i \mid X_{i-1})$$

Learning HMMs from fully observable data is easy



Learn 3 distributions:

$$P(X_1 = a) = \frac{\text{Count}(\#\text{ first letter was } a)}{N}$$

N := dataset size

select training data where letter was a

$$P(O_i = \text{pixel 17 is white} \mid X_i = a) = \frac{\text{Count}(\text{pixel 17 was white}, X_i = a)}{\text{any } \ldots \text{ in}}$$

$$P(X_i = a \mid X_{i-1} = b)$$

What if O is observed, but X is hidden

Log likelihood for HMMs when **X** is hidden

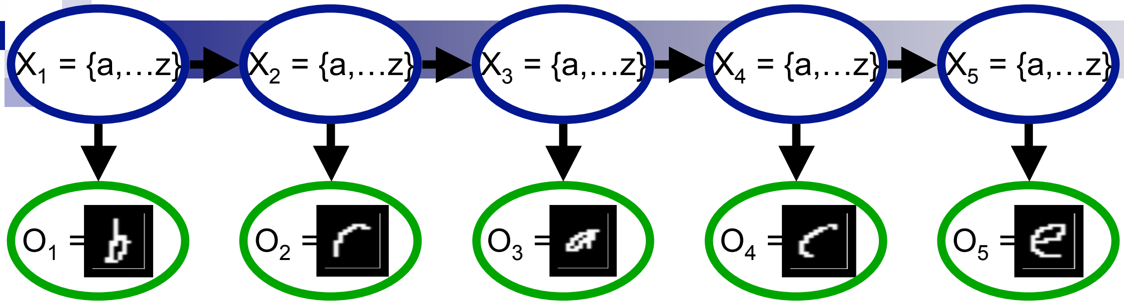
- Marginal likelihood – **O** is observed, **X** is missing
  - For simplicity of notation, training data consists of only one sequence:

$$
\begin{aligned}
\ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\
&= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta)
\end{aligned}
$$

  - If there were m sequences:

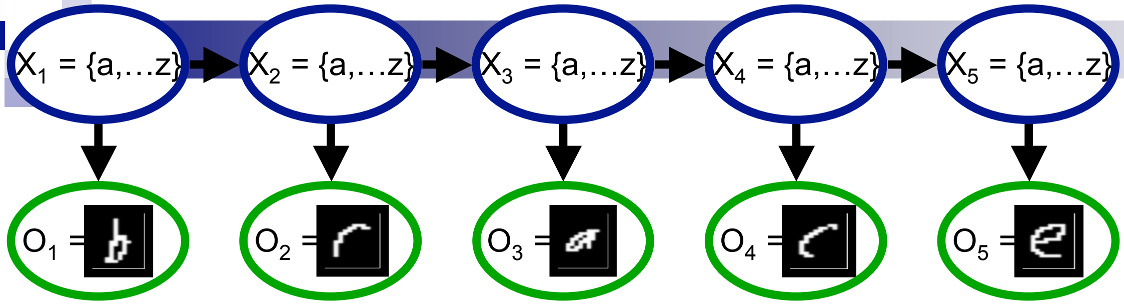
$$
\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o}^{(j)} \mid \theta)
$$

Computing Log likelihood for HMMs when **X** is hidden



$$\begin{aligned} \ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\ &= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta) \end{aligned}$$

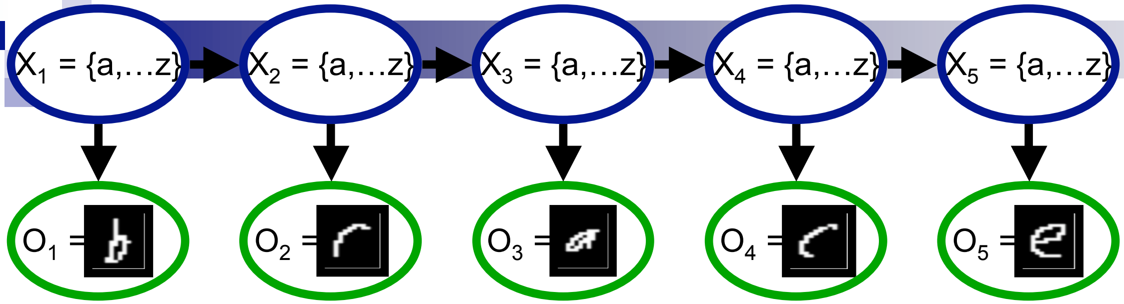
Computing Log likelihood for HMMs when X is hidden – variable elimination



- Can compute efficiently with variable elimination:

$$\begin{aligned}\ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\ &= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta)\end{aligned}$$

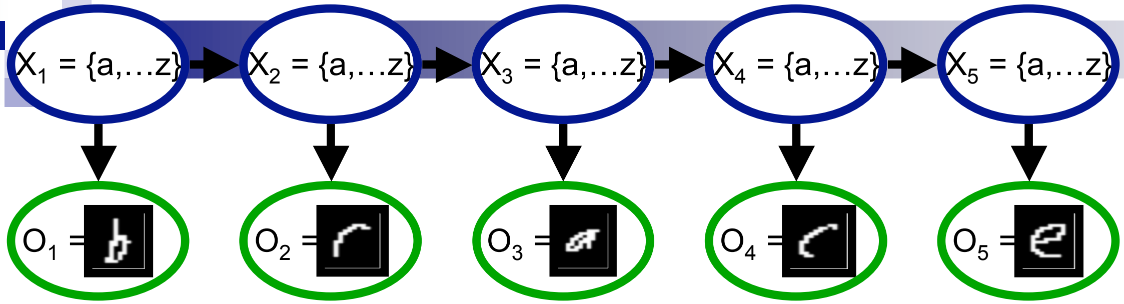
EM for HMMs when **X** is hidden



- E-step: Use inference (forwards-backwards algorithm)

- M-step: Recompute parameters with weighted data

E-step

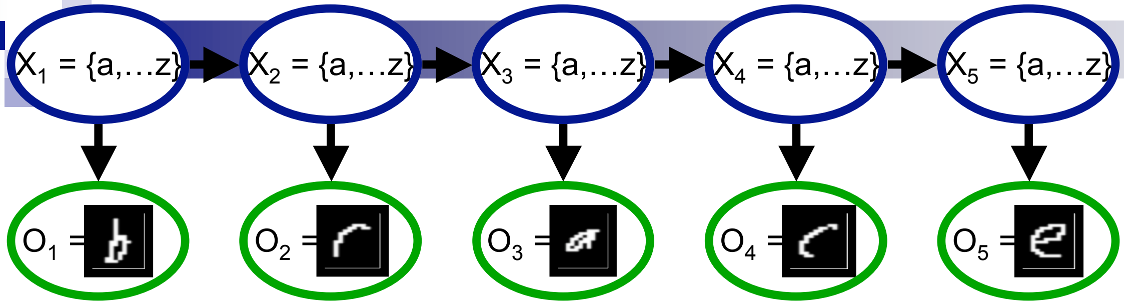


- E-step computes probability of hidden vars **x** given **o**

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \;=\; P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$

- Will correspond to inference
  - use forward-backward algorithm!

The M-step



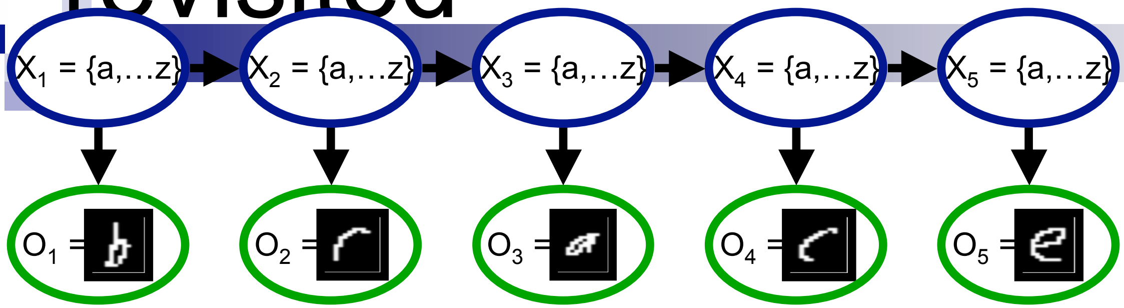
- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{\mathbf{x}} Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \log P(\mathbf{x}, \mathbf{o} \mid \theta)$$

- Use expected counts instead of counts:
  - If learning requires Count(**x**,**o**)
  - Use $E_{Q(t+1)}$[Count(**x**,**o**)]

Decomposition of likelihood revisited

$$P(X_1)$$
$$P(O_i \mid X_i)$$
$$P(X_i \mid X_{i-1})$$



- Likelihood optimization decomposes:

$$\max_\theta \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log P(\mathbf{x}, \mathbf{o} \mid \theta) =$$

$$\max_\theta \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log P(x_1 \mid \theta_{X_1}) P(o_1 \mid x_1, \theta_{O|X}) \prod_{t=2}^{n} P(x_t \mid x_{t-1}, \theta_{X_t|X_{t-1}}) P(o_t \mid x_t, \theta_{O|X})$$

Starting state probability $P(X_1)$

- Using expected counts
  - $P(X_1{=}a) = \theta_{X1=a}$

$$\max_{\theta_{X_1}} \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log P(x_1 \mid \theta_{X_1})$$

$$\boxed{\theta_{X_1=a} = \frac{\sum_{j=1}^{m} Q(X_1 = a \mid \mathbf{o}^{(j)})}{m}}$$

Transition probability $P(X_t|X_{t-1})$

- Using expected counts
  - $P(X_t=a|X_{t-1}=b) = \theta_{Xt=a|Xt-1=b}$

$$\max_{\theta_{X_t|X_{t-1}}} \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log \prod_{t=2}^{n} P(x_t \mid x_{t-1}, \theta_{X_t|X_{t-1}})$$

$$\boxed{\theta_{X_t=a|X_{t-1}=b} = \frac{\sum_{j=1}^{m} \sum_{t=2}^{n} Q(X_t = a, X_{t-1} = b \mid \mathbf{o}^{(j)})}{\sum_{j=1}^{m} \sum_{t=2}^{n} \sum_{i=1}^{k} Q(X_t = i, X_{t-1} = b \mid \mathbf{o}^{(j)})}}$$

Observation probability $P(O_t|X_t)$

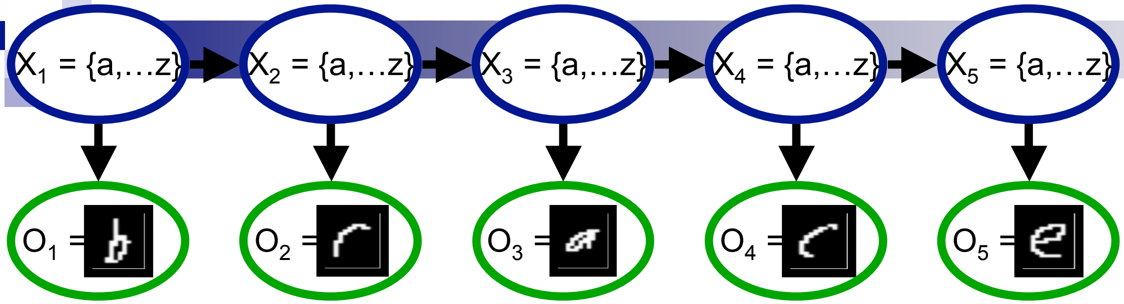
- Using expected counts
  - $P(O_t=a|X_t=b) = \theta_{Ot=a|Xt=b}$

$$\max_{\theta_{O|X}} \sum_{\mathbf{x}} Q(\mathbf{x} \mid \mathbf{o}) \log \prod_{t=1}^{n} P(o_t \mid x_t, \theta_{O|X})$$

$$\boxed{\theta_{O_t=a|X_t=b} = \frac{\sum_{j=1}^{m}\sum_{t=1}^{n} \delta(\mathbf{o}_t^{(j)} = a) Q(X_t = b \mid \mathbf{o}^{(j)})}{\sum_{j=1}^{m}\sum_{t=1}^{n} Q(X_t = b \mid \mathbf{o}^{(j)})}}$$

E-step revisited

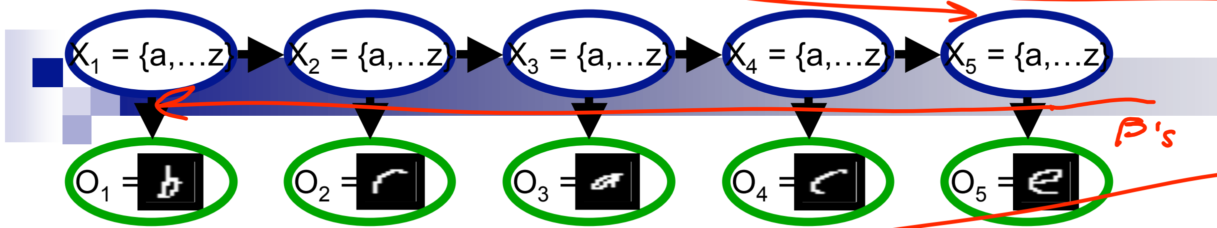
$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \;=\; P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$



- E-step computes probability of hidden vars **x** given **o**
- Must compute:
  - Q(x_t=a|**o**) – marginal probability of each position
  - Q(x_{t+1}=a,x_t=b|**o**) – joint distribution between pairs of positions

The forwards-backwards algorithm

α's



β's

$$P(X_i \mid o_{1..n})$$

drop O_1 in notation

- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$
- For i = 2 to n
  - Generate a forwards factor by eliminating X_{i-1}

sum out previous var prob obs   transition prob

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

$\alpha_5(a)$
$\alpha_5(b)$
$\vdots$
$\alpha_5(z)$

$\alpha_n(X_n)$ normalized $= P(X_n \mid O_{1:n})$

$\beta_1(X_1)\alpha_1(X_1)$ normalized $= P(X_1 \mid O_{1:n})$

- Initialization: $\beta_n(X_n) = 1$
- For i = n-1 to 1
  - Generate a backwards factor by eliminating X_{i+1}

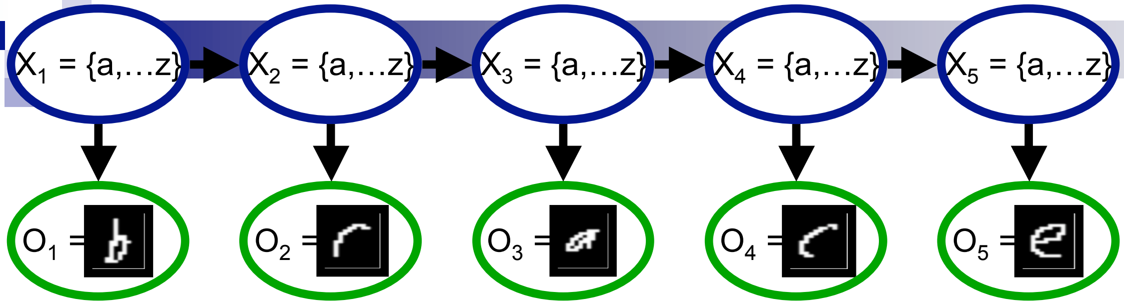
$\forall x_i$

$$\beta_i(\underset{=x_i}{X_i}) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid \underset{=x_i}{X_i})\beta_{i+1}(x_{i+1})$$

- 8 i, probability is: $\boxed{P(X_i \mid o_{1..n}) \propto \alpha_i(X_i)\beta_i(X_i)}$

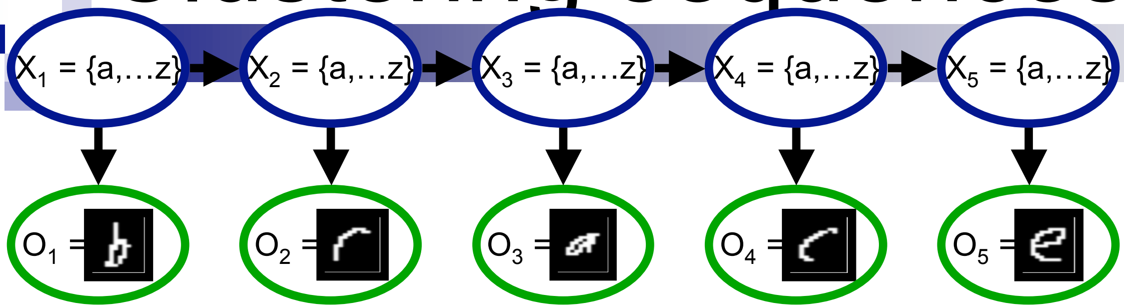
E-step revisited

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \;=\; P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$



- E-step computes probability of hidden vars **x** given **o**
- Must compute:
  - $Q(x_t=a|\mathbf{o})$ – marginal probability of each position
    - Just forwards-backwards!
  - $Q(x_{t+1}=a, x_t=b|\mathbf{o})$ – joint distribution between pairs of positions
    - Homework! ☺

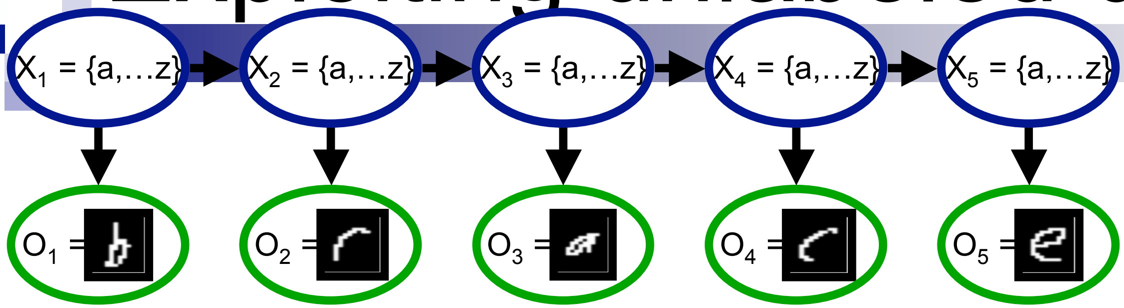
What can you do with EM for HMMs? 1 – Clustering sequences



Independent clustering:

Sequence clustering:

What can you do with EM for HMMs? 2 – Exploiting unlabeled data



- Labeling data is hard work ! save (graduate student) time by using both labeled and unlabeled data
  - Labeled data:
    - <X="brace",O= >
  - Unlabeled data:
    - <X=?????,O= >

Exploiting unlabeled data in clustering

- A few data points are labeled
  - <x,o>
- Most points are unlabeled
  - <?,o>
- In the E-step of EM:
  - If i'th point is unlabeled:
    - compute $Q(X|o_i)$ as usual
  - If i'th point is labeled:
    - set $Q(X=x|o_i)=1$ and $Q(X\neq x|o_i)=0$
- M-step as usual

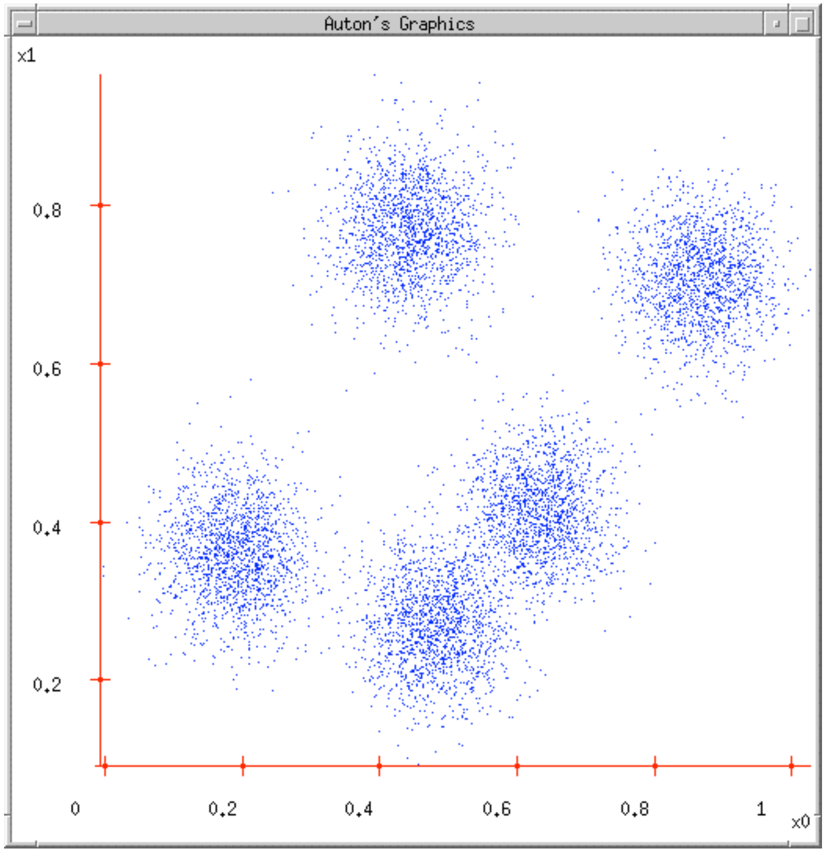
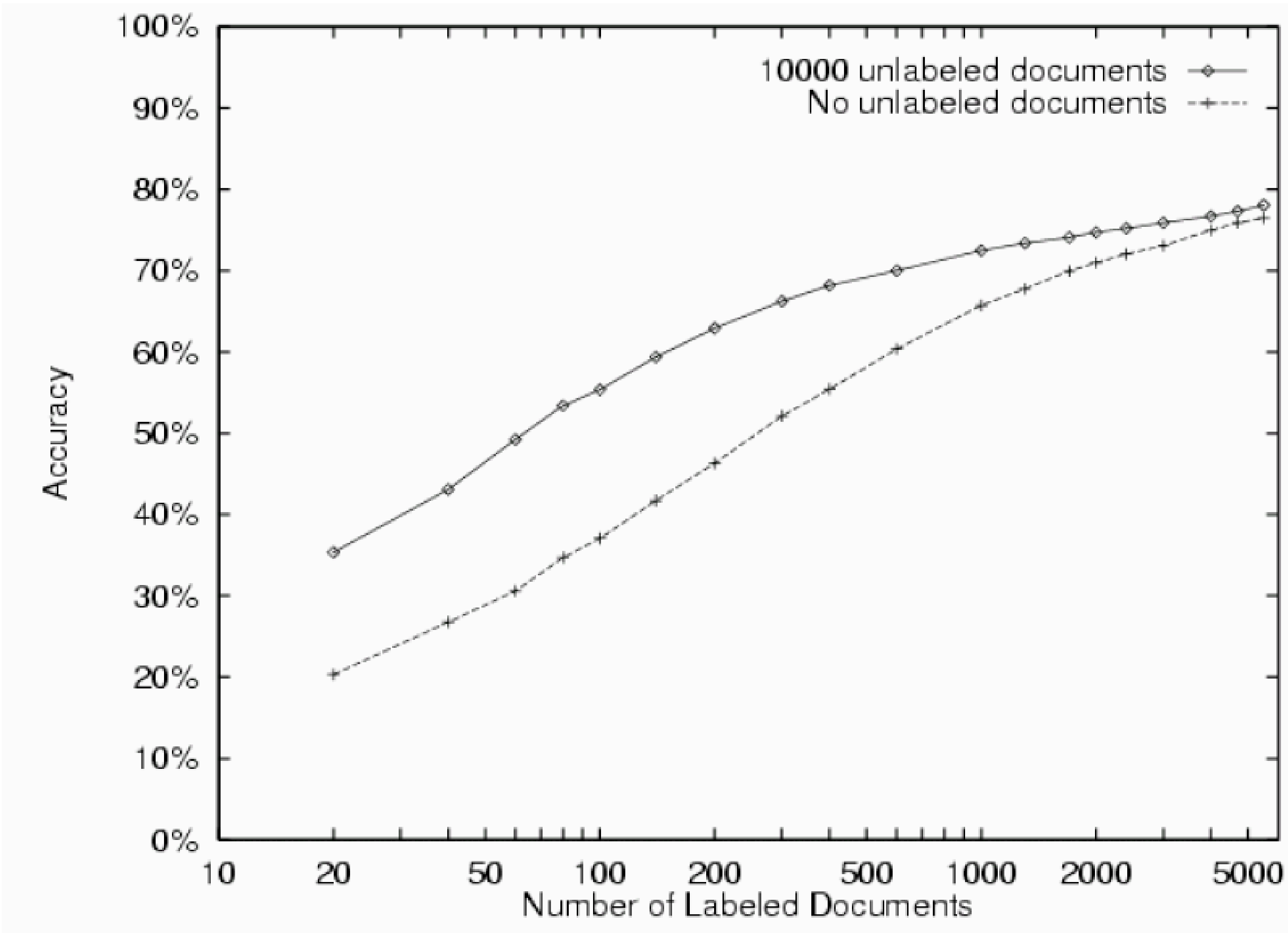


Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

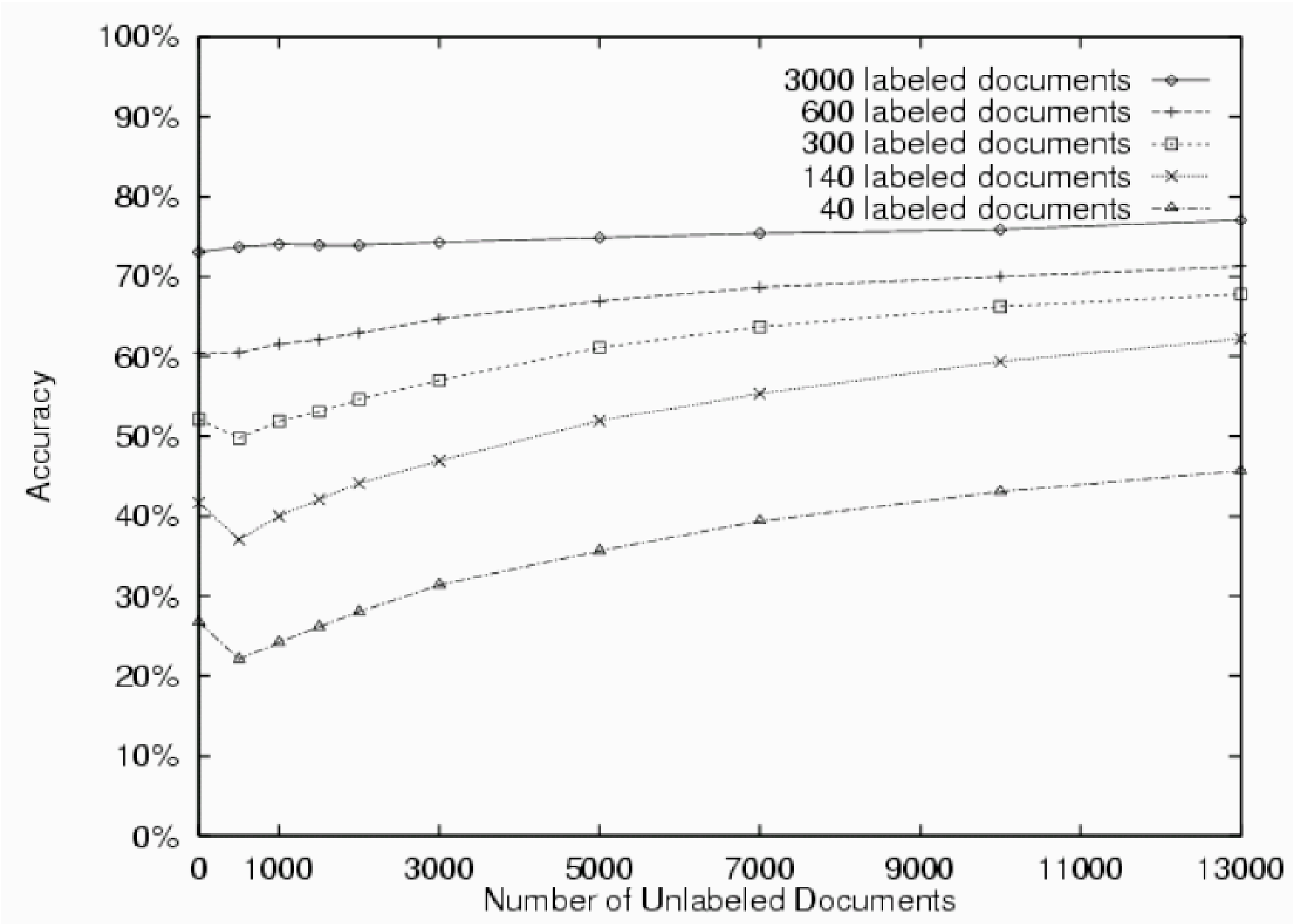
Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^{\star}$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^{\star}$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	DDam
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

Using one labeled example per class

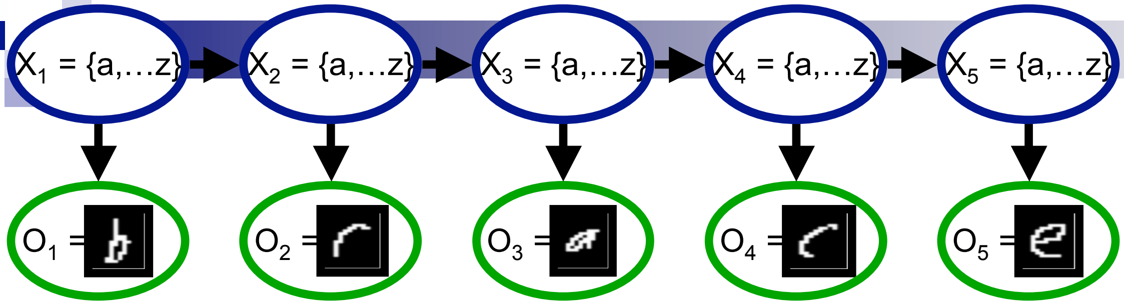
20 Newsgroups data – advantage of adding unlabeled data



20 Newsgroups data – Effect of additional unlabeled data



Exploiting unlabeled data in HMMs



- A few data points are labeled
  - <x,o>
- Most points are unlabeled
  - <?,o>
- In the E-step of EM:
  - If i'th point is unlabeled:
    - compute $Q(X|o_i)$ as usual
  - If i'th point is labeled:
    - set $Q(X=x|o_i)=1$ and $Q(X\neq x|o_i)=0$
- M-step as usual
  - Speed up by remembering counts for labeled data

What you need to know

- Baum-Welch = EM for HMMs
- E-step:
  - Inference using forwards-backwards
- M-step:
  - Use weighted counts
- Exploiting unlabeled data:
  - Some unlabeled data can help classification
  - Small change to EM algorithm
    - In E-step, only use inference for unlabeled data

Acknowledgements

- Experiments combining labeled and unlabeled data provided by Tom Mitchell